基于统计和特征相结合的查询纠错方法研究*

段建勇　关晓龙

(北方工业大学计算机学院　北京 100144)

摘要: 【目的】提高搜索引擎查询纠错过程中的准确率和召回率，改善用户的检索体验。【方法】提出一种基于统计和特征相结合的查询纠错模型，建立混淆集生成模型，将用户输入的查询关键字生成其对应的混淆集；建立混淆集排序模型，对混淆集中的词条进行排序，选出混淆集中最佳的词条与用户输入的查询关键字对照，以此达到查错纠错的目的。【结果】实验结果证明该模型在搜索引擎查询时具有较好的效果，测试集在 110k 时的准确率和召回率分别达到 92.2%和 95%，相对于 N-gram 纠错模型准确率和召回率分别提高 13.6%和 8.3%。【局限】该模型中混淆集的生成规则有限、模型的训练需要大量的计算。【结论】本模型能够提高搜索引擎查询的准确率及效率，改善用户的检索体验。

关键词: 查询纠错　混淆集　N-gram 模型　N-gram 相似度　编辑距离　点击词频

分类号: TP391　G35

1　引　言

随着互联网技术的不断进步和创新，人们对搜索引擎在查询、检索过程中的准确性和方便性提出更高的要求，这些需求对搜索引擎在查询纠错方面的技术也提出更高的挑战。对用户查询意图的识别研究[1]发现，用户在使用搜索引擎查询时，目标往往不是非常明确或者说是准确的，作为计算机系统来说，如何正确识别用户的查询、检索条件，对输入有误的查询关键字自动检错并纠错，并给出用户满意的查询结果成为搜索引擎查询技术研究的重要方面。

本文针对搜索引擎查询纠错的过程和方法进行研究，提出基于统计和语言特征相结合的查询纠错方法，建立模型并通过实验验证了该方法在搜索引擎查询纠错过程中的有效性，提高了搜索引擎的容错能力和易用性，同时也改善了用户的搜索体验。

2　研究现状

国外对于拼写纠错技术研究早于国内，英文文本的勘校中，不需要考虑分词问题，英文单词之间用空格分开，只需对单个词进行拼写检查，通常用编辑距离[2]计算词与词之间的相似度，再结合词在文本中的统计信息判断错误拼写，如 Senger 等[3]通过分析查询关键字的拼写错误以及错误的特征对药物信息系统的拼写错误进行纠正。

中文表达使用的是汉字，具有中文语言的特殊性。中文信息处理过程存在的同义词、同音词、多音字等问题常常会出现在中文的查错纠错中，使得中文的查询纠错变得错综复杂。目前中文查询纠错常见的方法有两种：基于字典的方法[4]和基于文本统计信息的方法[5]。基于字典的处理方法需要建立一个庞大的字典，应用字符串匹配的方式在字典中查询，查询纠

通讯作者：段建勇，ORCID：0000-0002-2244-3764，E-mail：duanjy@hotmail.com。

*本文系北京市社会科学基金项目"北京市公共危机事件在网络传播中的演化机制与模型研究"(项目编号:13SHC031)和国家自然科学基金项目"面向维基百科的多粒度一体化信息抽取方法研究"(项目编号:61103112)的研究成果之一。

错准确率很高，但是词典需要维护，随着网络和自然语言的飞速发展，新词、网络流行词汇层出不穷，仅仅依靠扩大词典的收录规模难以满足当前的查询纠错效率[4]。而基于文本统计信息的方法借助于大规模的语料库，从已有的大量实例中挖掘分析语言内在的关联关系及其特征，将其加入到统计模型中去，而不依赖于词典，也能取得较好的纠错效率。

现阶段对于查询纠错的研究，重点在利用网络数据和查询日志获取用户查询以及查询错误的规律特征，并将其使用到查询纠错中，如 Strohmaier 等[6]利用搜索引擎查询日志记录获取用户查询意图，Roy 等[7]通过对大量查询日志记录的严密分析，发现和理解用户的意图，取得了很好的效果。Subramaniam 等[8]结合编辑距离和基于查询日志的语言统计模型进行查询纠错。基于搜索引擎查询日志记录结合文本统计特征对用户的查询进行纠错，已成为现阶段搜索引擎查询纠错研究的重要方面。

3 主要方法

本文提出一种基于统计和特征相结合的查询纠错模型，建立混淆集生成模型对用户输入的查询串建模，并生成混淆集[9]。混淆集生成模型认为用户的所有输入是不可信的，但不是无用的，用户的目标查询串(用户真实意图的查询[9]串)可视为输入串经过混淆集生成模型而得到的，即用户的输入串经过混淆集生成模型后的混淆集中包含目标查询串。建立混淆集排序模型对混淆集中的候选串排序，筛选出最佳的候选串，并与原串比较得出纠错结果。这个过程将查错和纠错两个阶段合二为一，而且筛选出的最佳候选词条具有很高的正确率。整个纠错过程如图 1 所示:

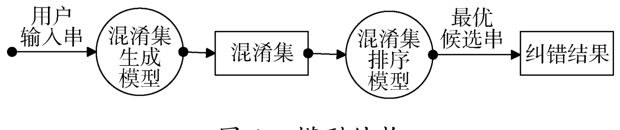


图 1 模型结构

在此过程中，有两个关键的步骤：依据用户输入查询串生成混淆集；对混淆集中的候选串进行综合评价并排序。生成的混淆集要保证用户的目标查询串包含其中，并且生成的混淆集的规模不能太大，即尽量避免不可能出现的错误词条，否则计算量太大而无法实现混淆集的排序。混淆集排序模型的建立是一个关键和重要的环节，此过程也是对混淆集中的候选串进行评价和选择的过程，候选串排序后选择其中评分最高的作为最优候选结果，与用户输入串比较得出纠错结果，此过程需要用到语言学、统计学、大量的数据挖掘和分析等知识，才能保证获取最优的候选结果。

3.1 混淆集生成模型

混淆集的生成模型的建立是整个纠错过程的关键，需要满足两个条件：尽量将所有可能出现的错误词条都包含到混淆集中；尽量不包含不可能出现的错误词条。满足第一个条件才能从混淆集中得到正确的纠错结果，提高纠错准确率；满足第二个条件可避免不可能出现的错误词条对纠错结果的干扰，避免大量的计算，提高纠错的效率。

用户输入的查询关键字并不可靠，所以不能直接以查询串为单位生成混淆集，先对每个输入查询串分词，针对每个分词产生各自的候选词集合，依据原输入查询串的分词结果将候选词交叉组合，形成混淆集。对搜索引擎查询日志记录分析可知，在用户输入的查询关键字中，93.15%的分词数目不超过 3[10]，这使得候选短语矩阵在可以接受和处理的范围之内。

用户输入的关键词有两个重要特点：错误都是局部的字词级别的；不同的输入方法具有相应的错误形式。即对于某个用户输入的词条，其出错的可能性集合即混淆集可以通过预先设定的规律有效生成。

王斯宇等[11]在基于 CSSCI 的文本自动校对系统的构建与实现中，采用基于混淆集和上下文特征的方法进行文本自动校对，对字、词语根据汉字的输入方式建立混淆集。而本文中的混淆集主要基于字音的方式生成分词的候选项，对每个分词的候选项交叉组合，形成候选项集合，即混淆集。具体过程如下：假定输入串为 $q = q_1q_2q_3\cdots q_n$，其中 q_i 表示第 i 个分词，对于 q_i 根据一定的规律生成候选项，这些规律参考文献[9]，主要包括以下方面：

(1) 多音字、同音词情况

用户在使用搜索引擎检索时，主要是靠输入法手动选择合适的字词，这个过程是没有音节的，而且拼音输入法的重码率很高，所以会出现同音字、多音字的选择，多音字情况例如：“大夫(医生)”和“大夫(古代官名)”；同音情况例如：输入“jishu”可能会出现“技

术”、“级数”、“奇数”等。这种因选择而导致的错误是最多的。

(2) 简写、缩写情况

现在的输入法为了方便用户，在用户输入查询关键字时，可能会出现汉字拼音的简写，例如用户输入“mdl”，可能代表的词语是“麦当劳”、“没电了”、“矛盾论”等。

(3) 音节歧义情况

在利用输入法输入查询关键字时，对于有些词语存在音节歧义的情况，例如：“xianren”可以分为“xian ren”即“仙人”、“线人”，也可以分为“xi an ren”即“西安人”。

(4) 近音词情况

近音词情况包含声母相似和韵母相似：声母最常混淆的有“s”和“sh”、“r”和“l”、“l”和“n”、“f”和“h”等；韵母最常混淆的有“ui”和“ei”、“an”和“ang”、“on”和“ong”等。

根据以上的候选项生成规律产生分词的候选项集合，交叉组合形成查询串的混淆集。混淆集生成过程如图2所示：

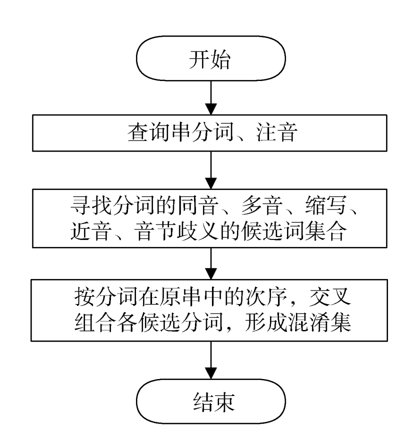


图2 混淆集生成过程

分词的候选词条交叉组合的过程如图3所示：

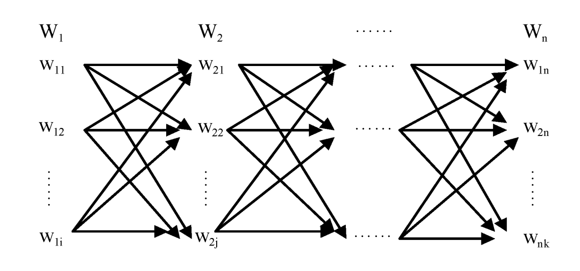


图3 混淆集生成过程

其中，字符串 $S = W_1W_2 \cdots W_n$，W_i 表示原输入串的第 i 个分词，w_{1i} 表示第一个分词的第 i 个候选项，w_{nk} 表示第 n 个分词的第 k 个候选项，依次交叉组合形成混淆集。

3.2 混淆集排序模型

从用户的输入串所生成的混淆集中选择最佳的候选词条，此过程其实是一个评价选择的过程。要使纠错结果最优，就要进行有效的评分，并对评分结果进行有效的排序。通过挖掘查询串的特征描述混淆集排序模型，从而得到有效的排序结果，选择最优的候选词条，得出纠错结果。所以构造有效的排序模型是一个非常重要和关键的环节，模型参数的确定也需要大规模的语料库训练得到，最终确定模型的形式。

为了很好地反映候选串的上下文特征，将自然语言处理中广泛应用的 N 元语法模型[5]引入到本文的特征中。目前对于互联网的大规模搜索引擎日志的用户行为分析也已经有一些研究成果，余慧佳等[10]选取搜狗搜索引擎一个月内的查询日志，就用户查询长度、查询频度、查询会话内的查询数目、以及用户点击行为进行分析，对用户的查询意图进行预测，可见查询日志记录的点击特征包含着用户的检索目的，这个特征也是很有价值的，因此本文也考虑进来。再考虑到以词形和编辑距离比较两个汉字串在形态上的相似程度[2]，即将 N 元语法模型、查询词点击率、词形相似度、编辑距离等因素以特征的形式建立候选项的排序模型并对候选词条排序，从而在候选集合中获得最优候选结果。

(1) N 元语法模型

N-gram 模型是自然语言处理中常用的算法模型，对于中文，又称为汉语语言模型(Chinese Language Model，CLM)。张仰森等[12]在中文的文本校对中使用 Bigram 模型，张仰森等[5]利用 trigram 和上下文依存关系分析来进行中文的自动查错，都取得了一定的效果。

从统计的角度看，自然语言中的一个句子 s，都是由一连串特定序列的词 $q_1q_2\cdots q_n$ 构成，根据链式规则，句子 s 出现的概率[5]为：

$$\begin{aligned}P(s) &= p(q_1)p(q_2 \mid q_1)p(q_3 \mid q_2q_1)\cdots p(q_n \mid q_{n-1}\cdots q_1) \\ &= \prod_{i=1}^{n} p(q_i \mid q_{i-1}\cdots q_1)\end{aligned} \tag{1}$$

可以认为对于每一个出现的词，其出现的概率取决于这个词前面的所有词。但是从计算上来看，由于

计算量太大而无法实现。N-gram 模型假定任意词的出现概率只和它前面的 $n-1$ 个词有关，即：

$$P(s)=\prod_{i=1}^{n}p(q_i\,|\,q_{i-n+1}\cdots q_{i-1}) \tag{2}$$

公式(2)是通过大量的语料统计和计算得出的，语料库的容量越大其频率值越接近其概率值，因此在大规模语料库的前提下，N-gram 模型可以表示为：

$$p(q_i\,|\,q_{i-1})=\frac{feq(q_iq_{i-1})}{feq(q_{i-1})} \tag{3}$$

其中，$feq(q_iq_{i-1})$ 表示 q_iq_{i-1} 在语料库中同现的频率，$feq(q_{i-1})$ 表示 q_{i-1} 在语料库中出现的频率。

由于语料库规模有限，许多合理的搭配关系在语料库中不一定出现，因此会出现数据稀疏现象(“零概率”问题)，通常在不扩大语料库规模的情况下，可以利用数据平滑技术进行调整，以消除模型参数为零的数据稀疏现象，使得模型参数的概率分布趋于均匀，提高模型整体的准确率。

目前已有很多数据平滑技术，如：Additive Smoothing 平滑、Add-one 平滑、Add-delta 平滑、Witten-Bell 平滑、Good-Turing 平滑、Jelinek-Mercer 平滑、Church-Gale 平滑、Katz 平滑等。本文应用 Additive Smoothing 平滑技术[9-10]，其计算方法如下：

$$P_{additive}(q_n\,|\,q_{n-k+1}\cdots q_{n-1})=\frac{\delta+feq(q_{n-k+1}\cdots q_n)}{\delta\,|\,V\,|+feq(q_{n-k+1}\cdots q_{n-1})} \tag{4}$$

其中 $0\leqslant\delta\leqslant1$，V 表示语料库中不同词的总数。对于二元语法模型，本文取 $\delta=1$，最终的二元语法模型计算公式为：

$$P_{additive}(q_i\,|\,q_{i-1})=\frac{1+feq(q_i,q_{i-1})}{|\,V\,|+feq(q_{i-1})} \tag{5}$$

(2) 查询词点击率

点击记录特征[10]可以衡量某个候选串的查询频度，查询频度是指在一段时间内，该查询词被提交的总次数。查询频次可以作为一个重要的启发来了解用户的搜索行为。Chen 等[13]利用日志的点击记录分析用户的偏好，从而提高查询纠错效率。万飞等[14]利用日志记录的点击率研究用户搜索行为，预测用户潜在需求。本文采用的点击记录是所用日志库中的查询词频。由于对输入串进行分词，所以对于多个词构成的候选串，取候选串各词频的均值，计算方法如下。

$$P_{CTR}(s)=p_{CTR}(q_1q_2\cdots q_n)=\frac{1}{n}\sum_{i=1}^{n}p_{CTR}(q_i) \tag{6}$$

(3) N-gram 相似度

本文需要计算用户输出查询串和各候选串在形态上的相似性，发现 N-gram 相似度[15]可以很好地解决这个问题。N-gram 相似度是指：利用 N-gram 思想，将词和词的相似度组合成 N 元词的 N-gram 相似度，进而通过计算不同长度的 N-gram 相似度得到用户输出查询串和各候选串的相似度。最经典的应用是机器翻译自动评测技术中的 BLUE 方法，本文参考此方法通过统计切分后的查询串和候选串的 N-gram 元组占候选串总 N-gram 元组的比例确定两个字符串的 N-gram 相似度。计算公式如下：

$$P_{sim_n\text{-}gram}(c,q)=\frac{\sum\limits_{n\text{-}gramc,n\text{-}gramq}count(n\text{-}gram)}{\sum\limits_{n\text{-}gramc}count(n\text{-}gram)} \tag{7}$$

其中分子表示查询串和候选串中能匹配的 N-gram 元组的数目，分母表示候选串中 N-gram 元组的数目。

(4) 编辑距离

编辑距离(Levenshtein Distance, LD)算法经常被用于字符串相似性问题的计算，在文本比较、信息处理等领域有着广泛的应用。编辑距离指两个字符串之间，由一个转换成另外一个所需要的最少的编辑操作次数。此处的编辑操作包括替换、插入、删除字符。

近年来，编辑距离算法在计算字符串相似度方面的改进取得了很大成就。Liang 等[16]将整条记录作为一个字符串，通过计算两个字符串的编辑距离判断两个字符串的相似程度。基于编辑距离计算两个字符串相似度的计算公式如下[2]：

$$p_1=1-\frac{ld}{m+n} \tag{8}$$

$$p_2=1-\frac{ld}{\max(m,n)} \tag{9}$$

其中，ld 是两字符串间的编辑距离，m, n 分别为两字符串的长度，但上述公式并不具普遍性。

赵作鹏等[17]提出一种基于改进的编辑距离相似度求解算法，其中包含编辑距离 LD，两个字符串的最长公共子串长度 LCS(s,t)，并考虑到两字符串比较时第

一次出现不匹配字符的位置δ。本文的编辑距离计算参照此改进算法，计算公式如下：

$$p_{Sim}(s,q)=\frac{lcs}{ld+lcs+\frac{L_m-\delta}{L_m}} \qquad (10)$$

其中s, q是比较的两个字符串，L_m是s串长度，ld为两字符串间的编辑距离，δ为两字符串第一次出现不匹配字符的位置。

(5) 混淆集排序模型的建立

对于生成的候选串，需要在排序后将原串与候选项集合中的最佳候选串比较，从而得出纠错结果。排序模型需要综合上述几个因素来综合给出评分，才能取得较好的排序效果。模型如下：

$$P_{overall-eval}(s,q)=\lambda_1 p(s)+\lambda_2 p_{CTR}(s)+\lambda_3 p_{sim_n\text{-}gram}(s,q)+\lambda_4 p_{Sim}(s,q) \qquad (11)$$

其中λ_1、λ_2、λ_3、λ_4表示N元语法模型、查询词点击率、词形相似度、编辑距离等特征的相应权重，并且$\lambda_1+\lambda_2+\lambda_3+\lambda_4=1$。

4 实验过程及结果分析

4.1 数据集

实验所用到的查询日志是从搜狗实验室获取的查询日志文件，通过去噪处理，包括排除特殊字符、错别字、无意义字符、重复记录的删除，从中提取具有代表性的记录，并对记录生成编号、注音，形成查询日志记录，记录数是50万条，结构如表1所示。实验用到的词典共收录词组 104 041 个，并且带有拼音(三字以上的形成简写)，词典文件的结构如表2所示。

表1 查询日志文件结构

词语编号	内容	拼音	带声调拼音	汉字数	词语数
1	[哄抢救灾物资]	hongqiangjiuzaiwuzi	hong1qiang2jiu4zai1wu4zi1	6	3
2	[汶川地震原因]	wenchuandizhenyuanyin	wen4chuan1di4zhen4yuan2yin1	6	3
3	[敬礼男孩心理障碍]	jinglinanhaixinlizhangai	jing4li3nan2hai2xin1li3zhang4ai4	8	4

表2 词典文件结构

编号	词语	拼音	带声调拼音	拼音简写
945	白面书生	baimianshusheng	bai2mian4shu1sheng1	bmss
978	白手起家	baishouqijia	bai2shou3qi3jia1	bsqj
3986	不可同日而语	buketongrieryu	bu4ke3tong2ri4er2yu3	bktrey

将词典中的词条按词语、拼音、带声调拼音、拼音简写(三字以上形成简写)、查询词频整理好后与日志库匹配，得到该词语在日志库中出现的总次数，查询词频的信息来自搜狗实验室提供的搜狗词频统计资料，这些词频作为实验中的查询词点击率，形成的语料库规模为106 246条记录，语料库文件结构如表3所示：

表3 语料库文件结构

词语编号	词语	拼音	带声调拼音	拼音简写	查询词频	日志中出现次数
203	安全理事会	anquanlishihui	an1quan2li3shi4hui4	aqlsh	0	73 040
204	安全门	anquanmen	an1quan2men2	aqm	7	226 164
205	报告文学	baogaowenxue	bao4gao4wen2xue2	bgwx	5	345 656

实验中用到的训练集是由查询日志记录而来，从日志记录的50万条中通过处理得到11万条训练记录，通过训练集训练得到混淆集排序模型的参数λ_1、λ_2、λ_3、λ_4。训练集结构如表 4 所示。测试集的生成是选取日志记录中的一些常用查询，在选取的词条中随机选取某词语，保证该词语出现在语料库中，然后生成该词语的候选项，替换词条中正确的词语，形成<错误词条，正确词条>的短语对。测试集文件结构如表5所示。

表4 训练集文件结构

词语编号	词语	纠错结果
61	林彪事件完整调查	1
124	极品家丁	0
127	公务员改革工资标准	1

(注：纠错结果中1表示纠错成功，0表示纠错未成功。)

表 5 测试集文件结构

编号	错误词条	错误词条拼音	正确词条	正确词条拼音	测试结果
1	姚命打架视频	yao2ming4da3jia4shi4pin2	姚明打架视频	yao2ming2da3jia4shi4pin2	1
2	陈水扁简历	chen2shui3bian3jian4li4	陈水扁建立	chen2shui3bian3jian3li4	0

4.2 评测指标

在对纠错系统的评价中，通常以召回率[18] (Recall，又称查全率)和准确率[18](Precision，又称查准率)作为评价标准，判断一个纠错模型的优劣。本实验中即采用这两个指标衡量模型的纠错效果。计算公式如下：

$$\text{Recall} = \frac{\text{纠错系统返回的不为空的词的个数}}{\text{关键词测试集中词的总数}} \tag{12}$$

$$\text{Precision} = \frac{\text{系统正确查出的错误词的个数}}{\text{关键词测试集中词的总数}} \tag{13}$$

4.3 实验过程及结果分析

利用查询日志记录(训练集)、语料库和词典，结合第 3 节中提出的纠错模型，得到纠错模型的最优参数，并利用模型对测试集数据测试验证纠错的效果。

利用语料库结合训练集得到模型的具体表达式，确定 λ_1、λ_2、λ_3、λ_4 的最优值，再用测试集验证模型在该组参数下的纠错效果。即读取测试集中的测试词条，对其进行分词操作，根据每个分词按照混淆集生成模型产生其候选项并交叉组合形成测试词条的混淆集，得到混淆集再结合混淆集排序模型，从 N-gram 特征、点击词频特征、词形相似度特征、编辑距离特征等方面对混淆集中的候选串进行评价、排序，计算方法见公式(1)。找出候选串中评分最高的候选串作为最优候选串与测试词条比较，若一致，则纠错成功，否则纠错失败。

整个实验过程可简单分两个步骤：

(1) 通过训练集得到模型的最优参数，如图 4 所示。

(2) 通过测试集验证模型的纠错效果，具体实验过程如图 5 所示。

实验得到模型的参数和纠错结果如表 6 所示，选取使得纠错的准确率最大的参数组作为模型的最优参数，即 λ_1=0.78、λ_2=0.20、λ_3=0.01、λ_4=0.01。

在不同规模的测试集下测试纠错模型的准确率和召回率，6 组测试集大小分别为 10k、30k、50k、70k、90k、110k，在不同规模的测试集下本文提出的纠错模型的准确率和召回率情况如图 6 所示。

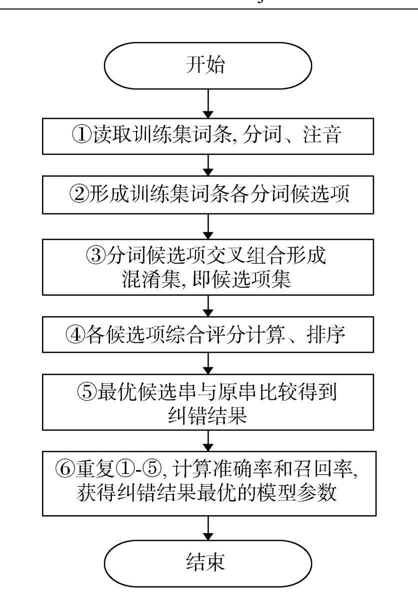


图 4 获取最优模型参数过程

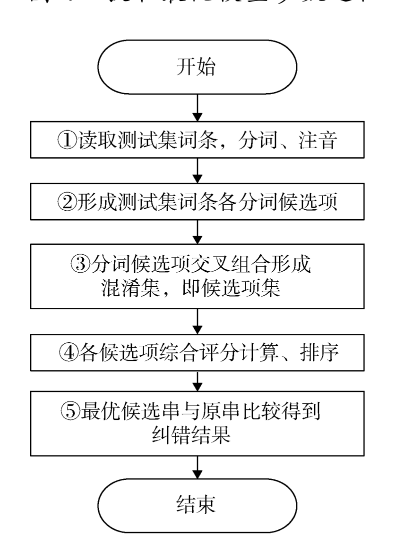


图 5 验证模型纠错效果过程

表 6 训练模型参数表

编号	λ_1	λ_2	λ_3	λ_4	召回率	准确率
1	0.01	0.01	0.01	0.97	93.17%	92.12%
2	0.01	0.01	0.02	0.96	93.28%	92.22%
3	0.01	0.01	0.03	0.95	93.21%	92.17%
…	0.50	0.01	0.01	0.48	93.25%	92.18%
…	0.97	0.01	0.01	0.01	93.17%	92.14%

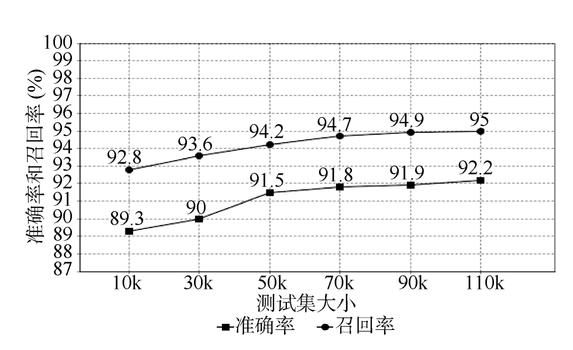


图 6　不同测试集下模型准确率和召回率

从图6的结果可以看出，本文提出的查询纠错模型所考虑的统计的特征信息能使纠错过程的准确率和召回率达到一定效果，因此该纠错模型是可行且有效的。

将本文提出的模型实验结果和单独考虑 N-gram 特征、N-gram 相似特征、编辑距离特征的情况做比较，即 $\lambda_1=1$ 、$\lambda_3=1$ 、$\lambda_4=1$，单独考虑各特征且在不同测试集下纠错的准确率如图 7 所示：

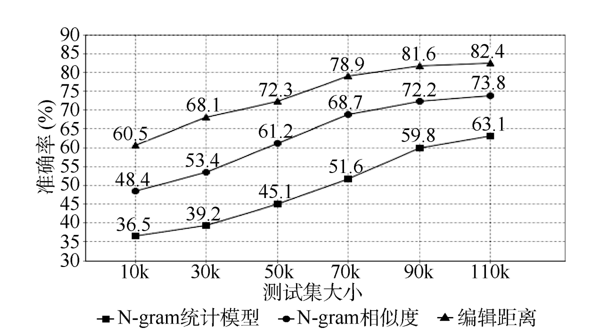


图 7　对比单独考虑各特征时的准确率

从图 7 可以看出纠错过程中只考虑单一的输入串的统计特征即只考虑N元语法模型、编辑距离、N-gram 相似度其纠错的准确率是偏低的。

陈智鹏等[19]提出通过分析上下文统计信息的方法，建立 N-gram 统计模型，实现搜索引擎中对查询关键字的自动检查和纠错。实验结果如图 8 所示：

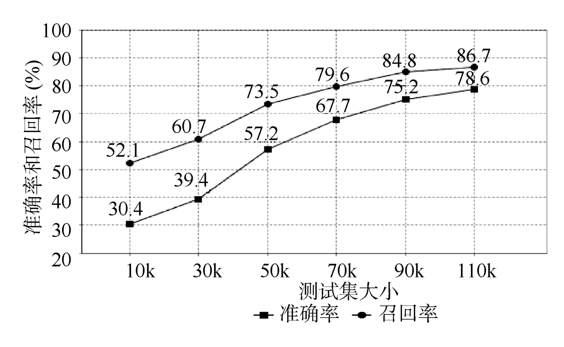


图 8　N-gram 统计模型的纠错准确率和召回率

从实验结果的统计信息可以得出以下结论：

(1) 通过图 6 和图 7，可以看出纠错过程中只考虑单一的输入串的统计特征即只考虑 N 元语法模型、编辑距离、N-gram 相似度其纠错的准确率是偏低的。

(2) 比较图 6 和图 8，可以得出如果纠错模型中包含的上下文统计信息比较多时，纠错的效果比较好。图 6 是本文方法的实验结果，图 8 是建立 N-gram 统计模型实现查询纠错，两者的效果差异很明显，本文提出的纠错模型较 N-gram 纠错模型，在测试集最大为 110k 时，纠错的准确率和召回率分别提高了 13.6%和 8.3%。

(3) 通过图 6、图 7、图 8 的比较还可以得出，本文提出的纠错模型是合理且有效的，基于各统计特征的结合，将查询串的各统计特征综合起来进行纠错，纠错的准确率和召回率能获得理想的数值且能保持稳定，并且随着测试集规模的增大，该模型可以获得的统计信息和特征信息就越多，纠错的准确率和召回率也随之提高。

(4) 通过对比图 7 和图 8 可以得出，实验中存在误差，比较两次实验结果误差范围仅仅在 4%-6%之间，在可以接受的范围之内。

5　结　语

本文提出一种基于统计和特征结合的查询纠错模型，通过对输入串的统计特征进行分析，结合 N 元语法模型、点击率、N-gram 相似度、编辑距离等，形成输入串的混淆集，结合特征对混淆集中的候选词条进行评价排序，将第一候选项与输入串比较得到纠错结果。通过实验可知，模型的准确率和召回率受语料库大小的影响，随着语料库的增大，模型能取得比较好的准确率和召回率。但是本文也存在一些不足：混淆集的生成规则有限，只考虑了 4 种情况；模型的训练需要大量的计算。为了提高模型的效率，在今后的研究中，需要对以上不足进行改进，使得模型的纠错准确率和效率得到进一步提升。

参考文献：

[1] 罗成，刘奕群，张敏，等．基于用户意图识别的查询推荐研究[J]．中文信息学报，2014，28(1)：64-72．(Luo Cheng, Liu Yiqun, Zhang Min, et al. Query Recommendation Based

on User Intent Recognition [J]. Journal of Chinese Information Processing, 2014, 28(1): 64-72.)

[2] 姜华，韩安琪，王美佳，等. 基于改进编辑距离的字符串相似度求解算法[J]. 计算机工程，2014，40(1)：222-227. (Jiang Hua, Han Anqi, Wang Meijia, et al. Solution Algorithm of String Similarity Based on Improved Levenshtein Distance [J]. Computer Engineering, 2014, 40(1): 222-227.)

[3] Senger C, Kaltschmidt J, Schmitt S P W, et al. Misspellings in Drug Information System Queries : Characteristics of Drug Name Spelling Errors and Strategies for Their Prevention [J]. International Journal of Medical Informatics, 2010, 79(12): 832-839.

[4] 胡晓青. 网络搜索引擎中文纠错功能实例剖析[J]. 图书情报工作网刊，2008(1)：1-6. (Hu Xiaoqing. The Examples Analysis of Chinese-Error Correction Function in Search Engines [J]. Library and Information Service Online, 2008(1): 1-6.)

[5] 张仰森，曹元大，俞士汶. 基于规则与统计相结合的中文文本自动查错模型与算法[J]. 中文信息学报，2006，20(4)：1-7, 55. (Zhang Yangsen, Cao Yuanda, Yu Shiwen. A Hybrid Model of Combining Rule-based and Statistics-based Approaches for Automatic Detecting Errors in Chinese Text [J]. Journal of Chinese Information Processing, 2006, 20(1): 1-7, 55.)

[6] Strohmaier M, Kroll M. Acquiring Knowledge About Human Goals from Search Query Logs [J]. Information Processing and Management, 2012, 48(1): 63-82.

[7] Roy R S, Katare R, Ganguly N, et al. Discovering and Understanding Word Level User Intent in Web Search Queries [J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2015, 30:22-38.

[8] Subramaniam L V, Roy S, Faruquie T A, et al. A Survey of Types of Text Noise and Techniques to Handle Noisy Text [C]. In: Proceedings of the 3rd Workshop on Analytics for Noisy Unstructured Text Data, Barcelona, Spain. New York, NY, USA: ACM, 2009: 115-122.

[9] 王永景. 面向文本识别流的自动校对算法研究[D]. 上海：上海交通大学，2008. (Wang Yongjing. The Research on the Automatic Proofreading Algorithm of Recognition Flow [D]. Shanghai: Shanghai Jiaotong University, 2008.)

[10] 余慧佳，刘奕群，张敏，等. 基于大规模日志分析的搜索引擎用户行为分析[J]. 中文信息学报，2007，21(1)：109-114. (Yu Huijia, Liu Yiqun, Zhang Min, et al. Research in Search Engine User Behavior Based on Log Analysis [J]. Journal of Chinese Information Processing, 2007, 21(1): 109-114.)

[11] 王斯宇，邵波. 基于 CSSCI 的文本自动校对系统的构建与实现[J]. 高校图书馆工作，2014，34(6)：50-54. (Wang Siyu, Shao Bo. The Construction and Implementation of Text Automatic Proofreading System [J]. Library Work in Colleges and Universities, 2014, 34(6): 50-54.)

[12] 张仰森，丁冰青. 基于二元接续关系检查的字词级自动查错方法[J]. 中文信息学报，2001，15(3)：36-43. (Zhang Yangsen, Ding Bingqing. Automatic Errors Detecting of Chinese Texts Based on the Bi-neighborship [J]. Journal of Chinese Information Processing, 2001, 15(3): 36-43.)

[13] Chen Q, Li M, Zhou M. Improving Query Spelling Correction Using Web Search Results [C]. In:Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Prague, Czech Republic. 2007: 181-189.

[14] 万飞，赵溪，梁循，等. 基于移动互联网日志的搜索引擎用户行为研究[J]. 中文信息学报，2014，28(2)：144-150. (Wan Fei, Zhao Xi, Liang Xun, et al. Search Behavior Study Based on the Mobile Search Log [J]. Journal of Chinese Information Processing, 2014, 28(2): 144-150.)

[15] 王金铨，梁茂成，俞洪亮. 基于 N-gram 和向量空间模型的语句相似度研究[J]. 现代外语，2007，30(4)：405-413. (Wang Jinquan, Liang Maocheng, Yu Hongliang. A Measure of Sentence Similarity Based on N-grams and Vector Space Model [J]. Modern Foreign Languages, 2007, 30(4): 405-413.)

[16] Liang J, Chen L, Mehrotra S. Efficient Record Linkage in Large Data Sets [C]. In: Proceedings of the 8th International Conference on Database System for Advanced Application. IEEE Computer Society, 2003: 137-146.

[17] 赵作鹏，尹志民，王潜平，等. 一种改进的编辑距离算法及其在数据处理中的应用[J]. 计算机应用，2009，29(2)：424-426. (Zhao Zuopeng, Yin Zhimin, Wang Qianping. An Improved Algorithm of Levenshtein Distance and Its Application in Data Processing [J]. Journal of Computer Applications, 2009, 29(2): 424-426.)

[18] 邵艳秋. 信息检索相关术语[J]. 术语标准化与信息技术，2009(4)：9-43. (Shao Yanqiu. Some Information Retrieval Terms [J]. Terminology Standardization and Information Technology, 2009(4): 9-43.).

[19] 陈智鹏，吕玉琴，刘华生，等. 基于 N-gram 统计模型的搜索引擎中文纠错[J]. 中国电子科学研究院学报，2009，14(3)：323-326. (Chen Zhipeng, Lv Yuqin, Liu Huasheng, et al. Chinese Spelling Correction in Search Engines Based on N-gram Model [J]. Journal of China Academy of Electronics and Information Technology, 2009, 14(3): 323-326.)

作者贡献声明:

段建勇: 提出研究思路, 设计研究方案, 研究方法的提出包括混淆集的生成模型和排序模型;
段建勇, 关晓龙: 进行实验, 实验数据分析, 论文起草及最终版本修订;
关晓龙: 采集、清洗和分析数据, 即训练集、测试集、语料库。

利益冲突声明:

所有作者声明不存在利益冲突关系。

支撑数据:

支撑数据见期刊网络版 http://www.infotech.ac.cn。

[1] 段建勇, 关晓龙. weblogex.xml. "搜狗实验室"提供的用户查询日志(SougouQ)迷你版.
[2] 段建勇, 关晓龙. dictionary.xml. 中文字典, 含有拼音及声调的汉字字典.
[3] 段建勇, 关晓龙. corpus.xml. 语料库, 包含模型计算所用的词频.
[4] 段建勇, 关晓龙. wordclick.xml. 分词点击率, 来自搜狗实验室.
[5] 段建勇, 关晓龙. trainwords.xml. 测试集数据.

收稿日期: 2015-08-03
收修改稿日期: 2015-10-09

Auto-Correction Search Model Based on Statistics and Characteristics

Duan Jianyong Guan Xiaolong
(College of Computer Science, North China University of Technology, Beijing 100144, China)

Abstract: [**Objective**] This study aims to improve the precision, recall and user experience of the search engine. [**Methods**] We proposed an automatic query correction model based on the statistics and characteristics. First, established a model to generate the confusion query set for the users' search terms, Then, created a ranking algorithm for the confusion set and chose the best match for the original queries. [**Results**] Our new model improved the search engine's performance. The precision and recall rates were 92.2% and 95% on a testing set of 110k, which were 13.6% and 8.3% higher than those of the N-gram model. [**Limitations**] Our model only generated four types of words for the confusion set, and the training process required a lot of computation. [**Conclusions**] The new model can improve the precision, recall and user experience of the search engine.

Keywords: Query correction Confusion sets N-gram model N-gram similarity Levenshtein Distance(LD) Frequent click rate